

WHAT IS CLAIMED IS:

1. A method of identifying a message source in a network, comprising:
 - receiving a method call from a client computer to invoke an object on a data server;
 - packaging the method call in a message to be sent from a client server to the data server via the network;
 - on the client server, identifying the client computer from an execution stack; and
 - transmitting the message to the data server.
2. The method of claim 1, further comprising:
 - on the client computer, generating the method call to invoke the object on the data server.
3. The method of claim 2, wherein transmitting the message to the data server transmits an identifier of an object on the client computer invoking the object on the data server along with the message.
4. The method of claim 3, wherein the identifier is stored in a header of the message.
5. The method of claim 3, wherein the identifier comprises a fully qualified class name.
6. The method of claim 1, wherein the message comprises a simple object access protocol (SOAP) message.

7. The method of claim 6, wherein packaging the method call in a message comprises building up a SOAP request.
8. The method of claim 7, wherein transmitting the message comprises implementing a SOAP application programming interface (API).
9. The method of claim 8, wherein the SOAP API comprises a messaging API.
10. The method of claim 2, further comprising:
displaying a Web service graphical component representing the object; and
displaying an interconnecting graphical component representing an associated interaction between the client computer and the data server.
11. A client server configured to transmit messages to a data server via a network, comprising:
a client computer interface configured to receive a method call from a client computer to invoke an object on the data server;
and
a data processing unit coupled to the client computer interface, the data processing unit being configured to:
package the method call in a message to be sent from the client server to the data server via the network;
identify the client computer from an execution stack;
and
transmit the message to the data server.

12. The client server of claim 11, wherein the message is transmitted along with an identifier of an object on the client computer invoking the object on the data server.
13. The client server of claim 12, wherein the identifier is stored in a header of the message.
14. The client server of claim 12, wherein the identifier comprises a fully qualified class name.
15. The client server of claim 11, wherein the message comprises a simple object access protocol (SOAP) message.
16. The client server of claim 15, wherein packaging the method call in a message comprises building up a SOAP request.
17. The client server of claim 16, wherein transmitting the message comprises implementing a SOAP application programming interface (API).
18. The client server of claim 17, wherein the SOAP API comprises a messaging API.
19. A program product comprising machine-readable program code for causing, when executed, a computer to graphically emulate a network including at least a client computer, a client server, and a data server, the program product graphically emulating the network performing method steps of:

on the client computer, generating a method call to invoke an object on the data server;
packaging the method call in a message to be sent from the client server to the data server via the network;
on the client server, identifying an identifier of an object on the client computer invoking the object on the data server the client computer from an execution stack;
and
transmitting the message to the data server.

20. The program product of claim 19, wherein the identifier comprises a fully qualified class name.
21. The program product of claim 19, wherein the message comprises a simple object access protocol (SOAP) message.
22. An apparatus configured to identify a message source in a network, comprising:
 - means for receiving a method call from a client computer to invoke an object on a data server;
 - means for packaging the method call in a message to be sent from a client server to the data server via the network;
 - means, on the client server, for identifying the client computer from an execution stack; and
 - means for transmitting the message to the data server.